

# DISSASSEMBLER REPORT

DASARI SRINITH  
CS21BTECH11015

## REPORT:

- The code for the Dissasembler, takes input of hex digits from a text file and outputs the RISC-V instructions into the console/terminal.
- I have used "vectors in cpp" for most of the cases , like to read input , for printing labels for branch and jump instructions.
- The first thing we have to do is to find what type of instruction is the hex instruction corresponds to.
- The for each type of instruction a function is written and is the opcode doesn't match to any instruction set then the output will be "invalid instruction".
- Then in each function , for finding what operation is given , we compare the funct7 or funct3 or both and then using if-else statement the required output is printed.
- For converting a binary string to decimal value ,used for immediate value or value of registers, two functions are written , one for signed and one for unsigned , as whenever we use a I-type instruction for most of the cases, we need to consider the signed value , so based on the operation required function is called.
- Reference for the instructions is taken from the RISC-V card.
- Then also for converting binary to hex , another function is also written which is useful in case of lui instruction
- And before printing any instruction , the code first calls a "label\_check" function, which checks if the instruction has to be labeled or not . It compares the instrcution number to a vector V0 , in which each element is a pair of two integers , one being the line for which the instruction has to be labeled , the second being the label number .
- This is updated if any valid branch/jump instruction is called , else for any invalid instruction , the output will be saying that the jump/branch is out of range.
- For testing my code , I have written a few instructions consisting of all the required instructions in the RIPPES simulator and then gave the corresponding hex instructions to my code and checked if the output is matching or not. Based on which I debugged most of my code.